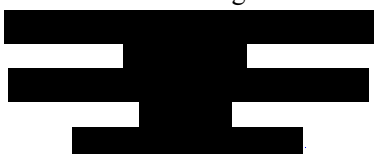
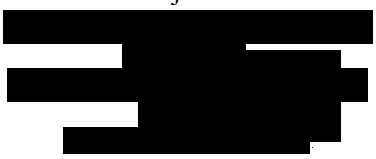
2010 Fourth International Conference on Network and System Security

Protecting from Attacking the Man-in-Middle in Wireless Sensor Networks with Elliptic Curve Cryptography Key Exchange

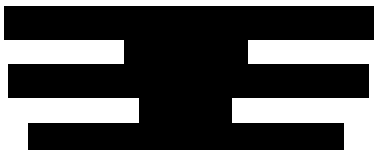
Xu Huang



Pritam Gajkumar Shah



Dharmendra Sharma



Abstract—**Today's security systems have been drawing great attentions as cryptographic algorithms have gained popularity due to the nature that make them suitable for use in constrained environment such as mobile sensor information applications, where computing resources and power availability are limited. Elliptic curve cryptography (ECC) is one of them, which requires less computational power, communication bandwidth, and memory in comparison with other cryptosystem. In particularly, in order to save pre-computing there is a trend for sensor networks to design a *sensor-group-leader* rather than every sensor node communicates to the end database, which indicated the needs to prevent from the man-in-the middle attacking. In this paper we first present an algorithm that we called "hidden generation point" ECC protocol to protecting the ECC key exchange system from the man-in-middle attacking in wireless sensor networks. Even though there are other ways to be investigated, which will published in other paper, the major contribution in this paper is showing the hidden generation point" works. Also it is noted that the agent technology provides a method for handling increasing software complexity and supporting rapid and accurate decision making. A multi-agent applying for key exchange is motioned even the further discussed will be presented in another paper as the major task of this paper is presenting "hidden generation point."**

Keywords- elliptic curve crytographic; public key; hiden generator point; man-in-middle attack.

I. INTRODUCTION

As the software applications have been permeating almost everywhere with ubiquitously computing, today's software applications are mainly characterized by their component-based structures which are usually heterogeneous and distributed. Agent technology provides a method for handling increasing software complexity and supporting rapid and accurate decision making. A number of different approaches have emerged as candidates for the agent architecture, and at the same time, dozens of environments for modelling, testing and finally implementing agent-based systems have been developed [1]. Software agent has been developed [2] and it is a well-known MAS (multi-agent system) development kit supporting a world-wide agreed agent standard. In this paper we are going to present MAS protect from wireless sensor network in ECC.

Security in communication system has become increasingly prominent and its key technology cryptography technology develops rapidly.

Wireless network has been experiencing an explosive growth in recent years and offering attractive flexibility to network operators and users.

Elliptic curve cryptography (ECC) has been known for public-key cryptography purposes [3], [4], which is independently introduced by Koblitz and Miller in 80's has attracted increasing attention in recent years due to its linear scalability, a small software footprint, low hardware implementation cost, low bandwidth requirement, and high device performance.

Normally the structure of an ECC operation involves three computational levels, namely scalar multiplication algorithm, point arithmetic and field arithmetic [5] mainly focus on improvements at the point arithmetic level to decreasing the time of ECC scalar multiplication. For point adding, a combination of projective and affine coordinates, i.e. mixed addition [6], has offered the efficient formulae. In the case of adding points in the same coordinate system, the required formula is more costly and is referred to as general addition. Recently, some approaches to compute faster scalar multiplications, such as double-base chain [7], ternary/binary method [6] have introduced tripling as a new point operation. Also there are many papers discussed the implementation over various situations [9]-[16]. Some of them raised different way to investigate the EEC such as multi-base system in ECC [17] and key exchange protocol in elliptic curve cryptography with no public point [10].

Recently, there is a trend for the sensor networks that the sensor group leaders rather than sensors communicate to the

Xu Huang is at the Faculty of Information Sciences and Engineering, University of Canberra, Australia.

Pritam Gajkumar Shah is a PhD candidate at the Faculty of Information Sciences and Engineering, University of Canberra, Australia

Dharmendra Sharma is at the Faculty of Information Sciences and Engineering, University of Canberra, Australia.

978-0-7695-4159-4/10 $26.00 © 2010 IEEE
DOI 10.1109/NSS.2010.15

end database as there is an effect way to save the storages for the "pre-computing data" in the sensor networks. However, the algorithms related to this trend will inherently create chance for an attacker to make a "man-in-middle" attack, which is shown in Figure 1.

Therefore there is necessary to think how to protect the networks from the attacks. In this paper we are going to present two ways to protect the network from man-in-middle attacks based on hidden generator point over elliptic curve cryptography for public keys.

The next section will discuss the general tradition elliptic curve cryptography with our focus that will be modified for our new protocols to protect the networks from the man-in-middle attacks. After this section, the two ways based on the hidden generator point will be investigated, with which we can see either way will carry on the protection works while the process of ECC is on the way to completed. The last section will give the conclusion of this paper.

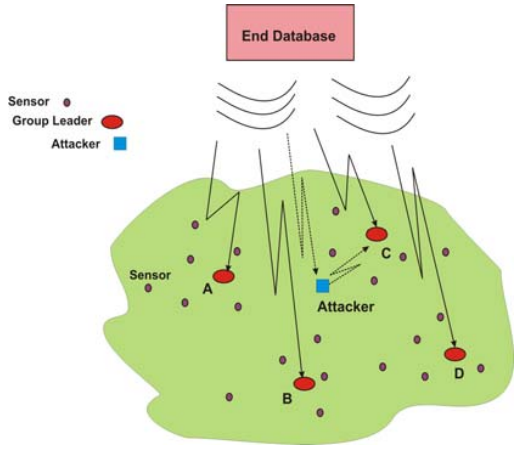


Figure 1: Schematic diagram shows possible attack made by "man-in-middle" in sensor network (in particular for the case with "group leader" communications.)

II. Traditional ECC Protocol

An elliptic curve is the set of solutions of an equation of the form can be shown as below:

$$y^2 + axy + by \quad x^3 + cx^2 + dx + e \tag{1}$$

where a, b, c, d, and e, are real numbers.

A special addition operation is defined over elliptic curves and this with the inclusion of a point O, called point at infinity. If three points are on a line intersecting an elliptic curve, then their sum is equal to this point at infinity O, which acts as the identity element for this addition operation. Sometimes the genera equation (1) can be referred as Weierstrass equation as shown in (2):

$$y^2 \quad x^3 + ax + b \tag{2}$$

If we wanted use a elliptic curve to be used for cryptography the necessary condition is the curve is not singular, i.e. the discriminant of polynomial $f(x) = x^3 + ax + b$:

$$4a^3 + 27b^2 \neq 0 \tag{3}$$

Figures 1 and 2 show the two elliptic curves are

$$y^2 \quad x^3 + 2x + 5 \tag{4}$$

and

$$y^2 \quad x^3 \quad 2x + 1 \tag{5}$$

We can see those two equations meet (3).

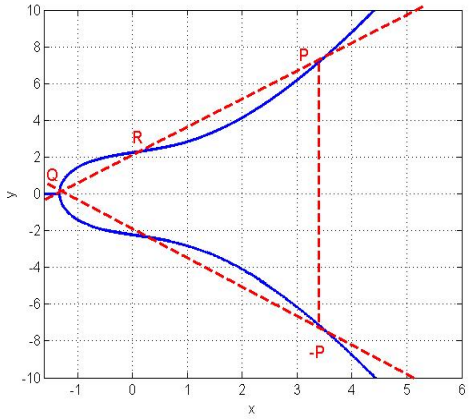


Figure 2: Elliptic curves equation (4)

An elliptic group over the Galois Field $E_p(a,b)$ is obtained by computing $x^3 + ax + b \bmod p$ for $0 \le x < p$. The constant a and b are non negative integers smaller than the prime number p and as here we used "mod p", so equation (3) should be read as:

$$4a^3 + 27b^2 \bmod p \neq 0 \tag{6}$$

For each value of x one needs to determine whether or not it is a quadratic residue. If it is the case, then there are two values in the elliptic group. If not, then the point is not in the elliptic $E_p(a,b)$ group.

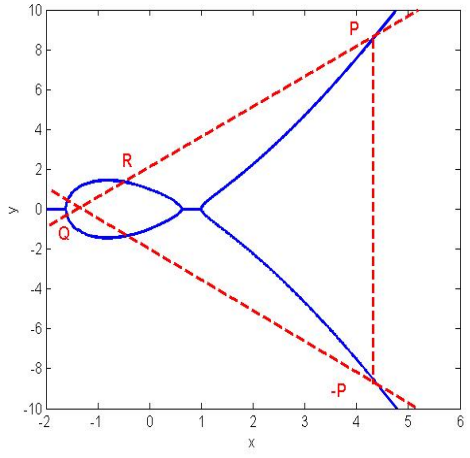


Figure 3: Elliptic curve equation (5)

When we fixed a prime number, p and then via the fixed constants a and b we have the Galois Field $E_p(a,b)$ group.

For example, let the points $P = (x_1, y_1)$ and Q (x_2,y_2) be in the elliptic group $E_p(a,b)$ group and O be the point at infinity. The rules for addition over the elliptic group $E_p(a,b)$ are :

(1) $P+O = O + P = P$

(2) If $x_2 = x_1$ and $y_2 = -y_1$, that is $P(x_1, y_1)$ and $Q = (x_2, y_2) = (x_1-y_1) = -P$, that is the case: $P+Q = O$.

(3) If $Q \neq -P$, then their sum $P+Q = (x_3, y_3)$ is given by ;

$$x_3 \quad \lambda^2 \quad x_1 \quad x_2 \bmod p$$

$$y_3 \quad \lambda(x_1 \quad x_3) \quad y_1 \bmod p \qquad (7)$$

where

$$\lambda \equiv \begin{cases} \dfrac{y_2 \quad y_1}{x_2 \quad x_1} & \text{if } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if } P \quad Q \end{cases} \qquad (8)$$

The classical Elliptic Curve Diffie Hellman (ECDH) scheme works as shown in the Figure 4. Initially Sensor Node A and Sensor Node B agree on a particular curve with base point P. They generate their public keys by multiplying P with their private keys namely K_A and K_B. After sharing public keys, they generate a shared secret key by multiplying public keys by their private keys. The secret key is $R = K_A*K_B*P$.

With the known values of Q_A, Q_B and P it is computationally intractable for an eavesdropper to calculate K_A and K_B which are the private keys of sensor node A and Sensor Node B. As a result, adversaries cannot figure out R which is the shared secret key.

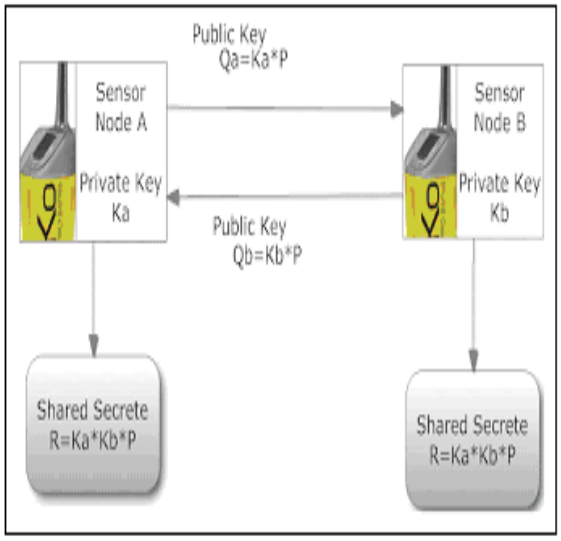


Figure 4: Elliptic Curve Diffie-Hellman (ECDH) protocol for WSN

In order to express our new protocol of the hidden generator point we, without losing generality, we use an example for the above description. Let's have $p = 23$ and $a = 1$ and $b = 1$, i.e. the equation becomes: $y^2 = x^3 + x + 1 \bmod 23$. We have $4a^3 + 27b^2 \bmod 23 = 8 \neq 0$. Now we need to determine if y_2 is in the set of quadratic residues or not. The calculation results are shown below for the elliptic group $E_p(a,b) = E_{23}(1,1)$ which includes the point (4, 0) corresponding to the single value $y = 0$.

The elliptic curve cryptography can be used to encrypt plaintext messages, M, into ciphertexts. The plaintext message M is encoded into a point PM from the finite set of points in the elliptic group, $E_p(a,b)$. First step consists in choosing a generator point, $G \in E_p(a,b)$, such that the smallest value of n for which $nG = O$ is a very large prime number. Normally the traditional ECC protocol is let the elliptic group $E_p(a,b)$ and the generator point G be in public. The each user select a private key, say $n_A < n$ and compute the public key P_A as $P_A = n_A G$. Then, encrypt the message point P_M for the partner, say from Alice to Bob. So Alice (A) chose a random integer k and computes the ciphertext pair of points P_C using Bob's public key P_B:

$$P_C \quad [(kG), (P_M + kP_B)] \qquad (9)$$

Bob received the ciphertext pair of points, P_C then multiplies the first point, (kG) with his private key, n_B, and then adds the result to the second point in the ciphertext pair of points as shown below:

$$(P_M + kP_B) \quad [n_B(kG)] \quad P_M \qquad (10)$$

which is the plaintext point, corresponding to the plaintext message M. It is noted that only Bob can obtain retrieve the plantext information P_M by the private key n_B. The cryptographic strength of ECC lies in the difficulty for a cryptanalyst to determine the secret random number k from kP and P itself. The fast method to solve this problem is known as the elliptic curve logarithm problem (ECLP) [18].

III. Proposed Protecting from Man-inMiddle Attcking in ECC

It is clearly to see that ECC did not take care of the man-in-the middle attacks even ECC itself has its cryptographic strength as described above.

As above shown that the generator point G and elliptic group $E_p(a,b)$ are in public. Now let's have a closer look at the elliptic group $E_p(a,b)$. In our above example, we pick the prime number $p = 23$ (it is noted that this is only for explaining the new protocol, in real life the p is bigger than this), we have quadratic residues group $(p-1)/2 = 11$ and for this group the $E_p(a,b)$ can be shown as below:

$$E_{23}(1,1) = \begin{Bmatrix} (0,1) & (0,22) & (1,7) & (1,16) & (3,10) & (3,13) & (4,0) \\ (5,4) & (5,19) & (6,4) & (6,19) & (7,11) & (7,12) & (9,7) \\ (9,16) & (11,3) & (11,20) & (12,4) & (12,19) & (13,7) & (13,16) \\ (17,3) & (17,20) & (18,3) & (18,20) & (19,5) & (19,18) & \end{Bmatrix} \qquad (11)$$

As we described in above that any point sitting in equation (3) can be appointed as generator point "*G* ," in the traditional way (as in section II) the *G* is fixed and let it be in public. But now we are not going to do so. As the generator is hidden, there is no way to know which point is generator therefore the attacker cannot make the "man-in-middle" attack. Now we are going to show two ways to complete the ECC processing, namely (1) make protocol that has the common principle to work out the generator point, say from the distribution of elliptic $E_p(a,b)$ group; or by the new protocol to work out the P_M as shown below.

In order to make a common principle to work out the generator point for Alice and Bob, say we are going to use the distribution of elliptic $E_p(a,b)$ group, we need to check the what it looks like. The equation (11) is shown in Figure 5.

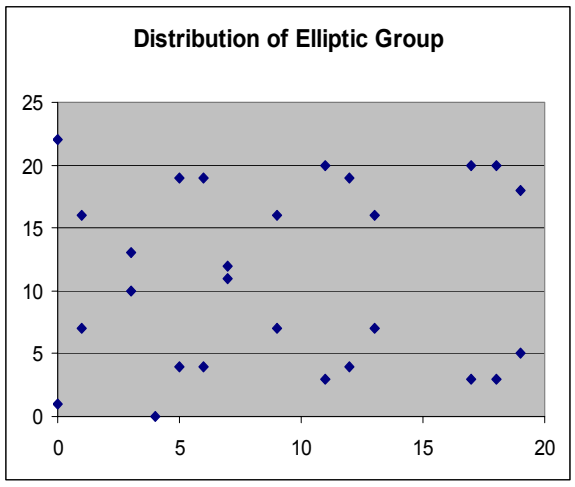


Figure 5: Distribution of Elliptic Group $E_{23}(1,1)$.

We now pick a generator point *G* by a character of the above distribution, say we pick the *G* when $G(a, b) \in E_p(a,b)$ with $a = \max\{a\}$ and $b = \max\{b\}$ (it is noted that other principle will apply). In this example, $G = (19, 18)$. Note that we put a = max {*a*} first, so choosing it $a = 19$ then choose $b = \max\{b\}$. The order is important, in this case it is not the $G = (18, 20)$. It is noted that there may have other ways to do the similar jobs as the discussed protocol did, for example we may make a "common agreement" for both sides that always take a particular point that is function of the distribution of $E_p(a,b)$. Here, we highlighted the function of the elliptic group distribution is because it would be hard to broken in comparison of fixed point. When the generator point fixed we can have the following processing as described in section II. In fact this way is more secure as the man (or women) has to do is decrypt the message from Bob re-encrypt it with Alice's key and he can monitor the communication without detection. This situation can be shown in Figure 6.

As the *G* is not at the public, the attack cannot work out *nG* as traditional way does, so even the attacker can monitor the communication but no way to understand and attack the communications. The yellow one in Figure 6 does not work for the new protocol and the real line works with a protecting.

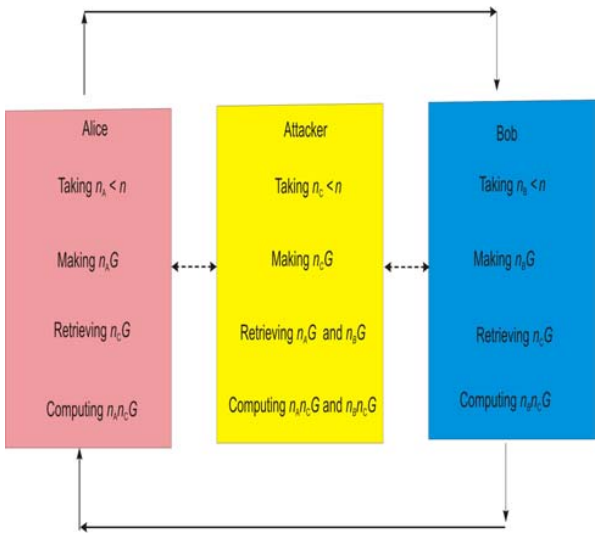


Figure 6: A new protocol protecting the man-in-the-middle attack.

This issue involves the user of a trusted "certificate authority" (CA). When is queried the CA and returns a digitally signed "certificate" that can be compared to one that has been transmitted by another means. In an authenticated key exchange based on the difficulty of the k^{th} root problem was described in section II. The way can be shown as Figure 5.

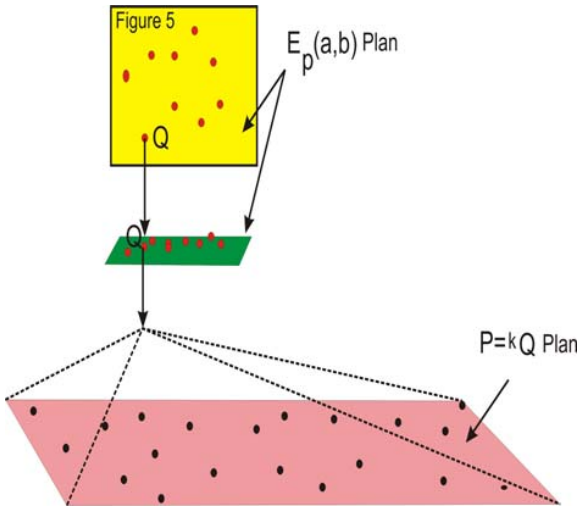


Figure 7: Block diagram for hidden generator point principle.

In Figure 7, the top plan is figure 5. When a generator is fixed by the distribution of the elliptic group then the $P = kQ$ plan formed.

Now let's turn to the 2nd way, i.e., a new protocol to get the hidden generator point done.

In the 2nd way, we need to face the case that Alice has no information about Bob's public key as traditional way does. Therefore if Alice would like to send a message to Bob, Alice

cannot use public key to make cryptography to the message Alice wanted to send. We may use, as an example, a protocol shown in Figure 8.

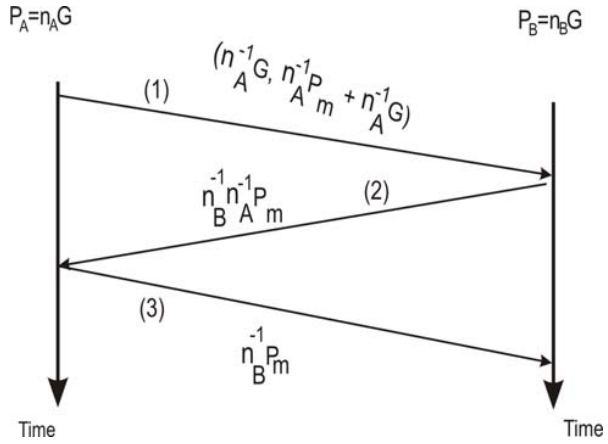


Figure 8: A protocol for the ECC with hidden generator point.

When Alice is going to send the message to Bob, Alice sends the pair of points P_C (as shown (1) in the figure) as below:

$$P_C \quad [(n_A{}^1G), (n_A{}^1P_M + n_A{}^1G)]$$

Here, $n^{-1}{}_A$ meets the equation: unity = $n^{-1}{}_A$ n_A, we still called $n^{-1}{}_A$ as private key for Alice but there is no need to worry about the public key as G is hidden at current situation. So either P_A or P_B is not really useful in this case. When Bob received P_C, he can operate as below:

$$n_A{}^1P_M \quad n_A{}^1P_M + n_A{}^1G \quad n_A{}^1G$$

Then, Bob can make P_D as below and sends it to Alice as shown the (2) in Figure 5.

$$P_D \quad n_A{}^1n_B{}^1P_M$$

When Alice received P_D, Alice can make P_E and sent it to Bob as shown (3) in the Figure 5.

$$P_E \quad (n_A)P_D \quad (n_A)(n_A{}^1n_B{}^1P_M) \quad n_B{}^1P_M$$

Then when Bob received P_E, Bob can obtain the message related P_M that sent from Alice by

$$P_M \quad n_Bn_B{}^1P_M$$

This can obtained only by Bob as no one has the private key that Bob has.

It is clear that the first way discussed above is less computing calculation in comparison with the second way but it is need the "common principle" or "common protocol" before the communication. If this common protocol is to be sent by communication network it will have a risk to be attacked or it will have to create a "safe way" to inform first then go ahead for the rest. For the second way, it is obviously it takes more time than that in traditional way, which is the price to pay for protecting communications from the man-in-middle attacks.

IV. Multiagent System Implementation of the ECC Public Key System

In the architecture, every agent is attached to a gateway federate. We can have several agents in a group attached to the same gateway federate or a single agent attached to a separate gateway federate. At the same time, different federates can also be put in the same machine or different machines. In order to enable agents to interact with their environment, we develop three interfaces to play the role of interaction channels between the federate and the agent, namely, a *Rule Induction Agent*, which is handling the ECC cryptography rule as described in section III, a *Dynamic Analysis Agent*, which is looking after related coding and decoding, and a *User Interface Agent*, which is dealing with interface processing. Both of them are transferred via the Object-to-Agent (O2A) communication channel provided by the agent toolkit. The framework of the multi-agent can be shown in Figure 9. Our work is concerned with the second area, namely to develop autonomous agents for representing entities in distributed simulations.

A multi-agent system comprised of multiple autonomous components needs to have certain characteristics (Jennings, Sycara, & Wooldridge, 1998; Roberto, 1999):

- each agent has incomplete capabilities to solve a problem;
- there is no global system control;
- data is decentralized; and
- computation is asynchronous.

That is, combining multiple agents in a framework presents a useful software engineering paradigm where problem-solving components are described as individual agents pursuing high-level goals.

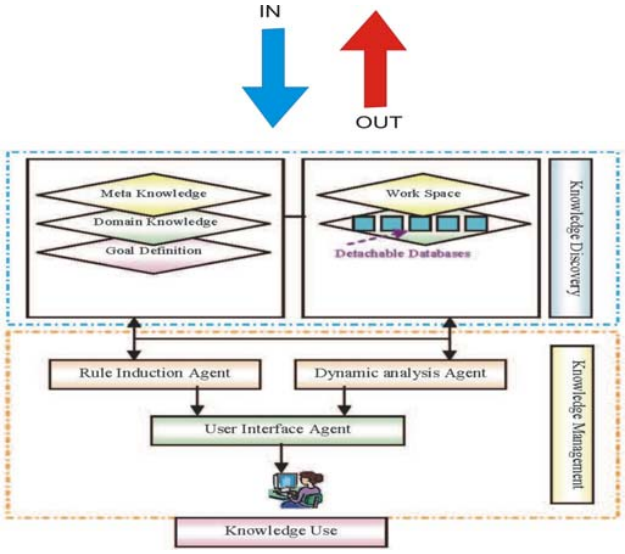


Figure 9: Multiagent framework for ECC

It is noted that there are two ways, i.e. one is the date out and another is date in, so it is easily to communicate between two end systems, which is looking after by the agents. With multi-agent the whole system is more effective and efficiency for the ECC public key system, which will be discussed in other paper.

V. CONCLUSION

As security issue has been paid more attentions. In recent years some cryptographic algorithms have obtained popularity due to properties that make them suitable for use in constrained environment such as mobile information appliances, sensor networks, where computing resources and power availability are limited. Elliptic curve cryptography (ECC) is one of them. However, in the applications of ECC, in particular for the sensor networks there are always so-called man-in–middle attacks, in particular those networks are with very limited computing capacity and restricted power resources, which drew the researchers' attractions. In this paper we have presented two methods for protecting from man-in-middle attacks based on hidden generator point with ECC with multiagent system implementation. A effective and efficient system was designed and tested.

REFERENCES

[1] Logan, B., Theodoropoulos, G.: The distributed simulation of multi-agent systems. Proceedings of IEEE **89** (2001) 174.186

[2] Eiter, T., Mascardi, V.: Comparing Environments for Developing Software Agents. INFSYS Research Report 1843-01-02, Tech. Univ. Wien, Austria (2001)

[3] V.S. Miller, "uses of elliptic curves in cryptography," in Advances in Cryptology, CRYPTO'85, ser . Lecture Notes in Computer Science, vol. 218, Springer, 1986. pp. 417-428.

[4] Research Report 1843-01-02, Tech. Univ. Wien, Austria (2001)

[5] N. Koblitz, " Elliptic curve cryptosystems," Mathematics of Compution, vol. 48, no.177, pp.203-209, Jan 1987.

[6] D. Hakerson, A. Menezes, and S. Vanston, "Guide to Elliptic Curve Cryptography," Springer-Verlag, NY (2004).

[7] H. Cohen, A Miyaji and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," Lectures Notes in Computer Science, 1514, 51-65 (1998).

[8] V. Dimitrov V., L. Imbert, and P. K. Mishra, "Efficient and secure elliptic curve point multiplication using double-base chains," Lectures Notes in Computer Science, 3788, 59-78 (2005).

[9] M. Ciet, M. Joye, K. Lauter and P.L. Montgomery,"Trading inversions for multiplications in elliptic curve cryptography," Designs, Codes, and Cryptography, 39, 189-206 (2006).

[10] D. Bernstein, "High-speed diffie-hellman, part 2," presented at the INDOCRYPT'06 tutorial session, Dec. 11-13, Kolkata, India (2006).

[11] K. Kaabneh and H. Al-Bdour, "Key exchange protocol in elliptic curve cryptography with no public point," American Journal of Applied Sciences 2 (8): 1232-1235, 2005.

[12] J. Adikari, V. Dimitrov, and L. Imbert, "Hybrid binary-ternary joint sparse from and its application in ellipic curve cryptography," Cryptology ePrint Archive, Report 2008/285, 2008.

[13] Bangju Wang, Huanguo Zhang and Yuhua Wang, "An efficient elliptic curves scalar multiplication for wireless network," 2007 IFIP International Conference on Network and Parallel Computing-Workship, pp131.

[14] Shiwei Ma, Yuanling Hao, Zhongqiao Pan, and Hui Chen, "Fast implementation for modular inversion and scalar multiplication in the elliptic curve cryptography," 2008 Second International Symposium on Intelligent Information Technology Application, pp488.

[15] Michael Scott, "Optimal Irreducible Polynomials for GF(2^m) Arithmetic," Cryptology ePrint Archive, Report 2007/192, 2007.

[16] H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography-based access control in sensor networks," International Journal of Security and Networks, vol. 1, no.3/4, 2006.

[17] A. Liu, P. Kampanakis, and P. Ning, "TinyECC: Elliptic curve cryptography for sensor networks ," (version 10), november 2007.

[18] C. Doche, D. Kohel, and F. Sica, "Double-base number system for multiscalar multiplications," Cryptology ePrint Archive, Report 2008/288. 2008.

[19] W. Stallings, "Cryptography and Network Security: Principles and Practice," Prentice-Hall, Upper Saddle River, New-Jersey, second edition, 1999.